ZJU Summer 2024 Training

Contest 8 by Group B

memset0 zzw4257 Clovers water_tomato Hydroxythio

2024-07-15

Zhejiang University

Project Sneaking

给定一张带权有向图，问点 s 到点 t 的最短时间。在任何时刻，可以在边上走，也可以在某个点停留。

但有一个限制，对于每个点，在其 $l_i - r_i$ 时间内是有巡逻的，即不可以在这些时段进入该点，或是在该点等待到该时段（包括等待过程中经历了该时段）。

题解

我们记录每个点在 l_i 前到达的最短时间，记为 f_i，和每个点在 r_i 后到达的最短时间，记为 g_i。

则若存在一条从 u 到 v，权值为 w 的边，我们有以下的转移方程：

1. 若 $f_u + w \le l_v$，则 $f_v = \min(f_v, f_u + w)$.
2. 若 $l_u + w \ge r_v$，则 $g_v = \min(g_v, \max(r_v, f_u + w))$.
3. 若 $g_u + w \le l_v$，则 $f_v = \min(f_v, g_u + w)$.
4. 不论如何，都有 $g_v = \min(g_v, \max(r_v, g_u + w))$.

题解

然后我们做一个 Dijkstra，注意，每个点的 f 和 g 是需要分别独立进入优先队列的，即一个点会分别在 f 成为堆顶和 g 成为堆顶的时候更新一次。具体地，我们可以开一个两倍大小的 vis 数组，对于一个点 u，其 f 进入优先队列时我们推入 pair (f_u, u)，g 进入优先队列时我们推入 pair $(g_u, u + n)$。

这样，我们就保证了每个点在 f 确定和 g 确定后都能进行一次更新，也就覆盖了所有的可能情况。

Memba Out

维护序列 a_i，设 f_i 为从位置 i 开始，左右单调栈上不同元素个数。支持两种操作：

- 交换 a_x 和 a_{x+1}；
- 询问 $f_l + f_{l+1} + \cdots + f_r$。

$1 \leq n, q \leq 3 \times 10^5$。

题解

考虑只维护左单调栈怎么做：

一开始先求出答案，之后如果交换 a_x, a_{x+1} 不仿设 $a_x < a_{x+1}$，那么只会对 $[1, x-1]$ 中最右边的一段 $a_i < x$ 的答案产生 -1 的影响，和对 $[x+1, n]$ 中最左边的一段 $a_i < x$ 的答案产生 $+1$ 的影响。具体是哪一段可以通过线段树上二分找出来。

题解

考虑只维护左单调栈怎么做：

一开始先求出答案，之后如果交换 a_x, a_{x+1} 不仿设 $a_x < a_{x+1}$，那么只会对 $[1, x-1]$ 中最右边的一段 $a_i < x$ 的答案产生 -1 的影响，和对 $[x+1, n]$ 中最左边的一段 $a_i < x$ 的答案产生 $+1$ 的影响。具体是哪一段可以通过线段树上二分找出来。

现在就是既要统计左右单调栈长度，又要减掉他们重复的数字个数。假设重复出现的数字是 $a_l = a_r = x$，这要求 $l+1, l+2, \cdots, r-1$ 中每个数都 $< x$。容易发现，这样需要考虑的区间是 $O(n)$ 级别的。

题解

考虑只维护左单调栈怎么做：

一开始先求出答案，之后如果交换 a_x, a_{x+1} 不仿设 $a_x < a_{x+1}$，那么只会对 $[1, x-1]$ 中最右边的一段 $a_i < x$ 的答案产生 -1 的影响，和对 $[x+1, n]$ 中最左边的一段 $a_i < x$ 的答案产生 $+1$ 的影响。具体是哪一段可以通过线段树上二分找出来。

现在就是既要统计左右单调栈长度，又要减掉他们重复的数字个数。假设重复出现的数字是 $a_l = a_r = x$，这要求 $l+1, l+2, \cdots, r-1$ 中每个数都 $< x$。容易发现，这样需要考虑的区间是 $O(n)$ 级别的。

只需要写一个单点修改，线段树上二分的线段树和一个单点修改、区间加、区间求和的线段树即可。

Blocking Game

给定一张有向图，图上有 q 个终点（输入中给出），有一个 bot 在图上移动，Beryl 和 Sapphire 一次进行若干轮游戏，每一轮：

1. Beryl 选 k 条边，封锁之（仅对这一轮有效）
2. Sapphire 令 bot 沿着一条未封锁的边的走一步。

对于每个点作为起点的情况，都询问 Sapphire 能不能在 10^9 步内胜利（走到终点即胜利，无路可走即失败）。

题解

首先，所有终点一定是 S 点。我们可以建一张反图，然后将所有 S 点（终点）加入队列，接着类似拓扑排序地方式进行更新，若有大于 k 个 S 点在反图中有一条指向某个点的边，那么这个点显然也可以走到一个 S 点，我们就把这个点也变成 S 点，加入队列。

证明：我们考虑终态，对于一个 B 点，他一定没有大于 k 条通往 S 点的边，那么只要一直封锁这些边，bot 就永远无法到达 S 点。对于一个 S 点，不论怎么封锁，其一定能到达一个比他更早进队的 S 点，故最终一定能到达一个终点。

Pain Flow

有 n 个随机变量 $a_1, a_2, ..., a_n$，其中 $a_i \sim \mathcal{U}(l_i, r_i)$ 即服从 l_i 到 r_i 的均匀分布。特别的，当 $l_i = r_i$ 时，$a_i = l_i$。求逆序对的期望，对 $10^9 + 9$ 取模。

题解

考虑一个暴力：枚举一对$(i,j)(i<j)$，算这一对的贡献，记$b_i=r_i-l_i$，首先考虑$b_i,b_j\neq 0$的情况。不妨采用几何概型，假设我们有一个长b_i宽b_j的矩形，放到平面直角坐标系上，由 $x=l_i,x=r_i,y=l_j,y=r_j$ 四根直线围成，点 (x_0,y_0) 表示 a_i 取 x_0,b_i 取 y_0 这一个事件，那么所有基本事件就是矩形的面积 $b_i\times b_j$，会对逆序对产生贡献的点还要满足 $a_i>a_j$，也就是在直线 $y=x$ 下方的部分，拿去和这个矩形求交，得到的图形面积记为 s_{ij}，那么贡献就是 $\frac{s_{ij}}{b_ib_j}$。

题解

考虑一个暴力：枚举一对$(i,j)(i<j)$，算这一对的贡献，记$b_i=r_i-l_i$，首先考虑$b_i,b_j\neq 0$的情况。不妨采用几何概型，假设我们有一个长b_i宽b_j的矩形，放到平面直角坐标系上，由 $x=l_i,x=r_i,y=l_j,y=r_j$ 四根直线围成，点 (x_0,y_0) 表示 a_i 取 x_0,b_i 取 y_0 这一个事件，那么所有基本事件就是矩形的面积 $b_i\times b_j$，会对逆序对产生贡献的点还要满足 $a_i>a_j$，也就是在直线 $y=x$ 下方的部分，拿去和这个矩形求交，得到的图形面积记为 s_{ij}，那么贡献就是 $\frac{s_{ij}}{b_ib_j}$。

s_{ij} 可以 $O(1)$ 计算，这样我们得到一个 $O(n^2)$ 的算法。

题解

考虑优化这个暴力，不妨固定 j，同时计算所有可行的 i 的答案，也就是 $\sum_{i=1}^{j-1} \frac{s_{ij}}{b_i b_j}$。先把 b_j提出来，那么只需要考虑 $\frac{s_{ij}}{b_i}$,下面我们来研究 s_{ij} 的性质。

题解

考虑优化这个暴力，不妨固定 j，同时计算所有可行的 i 的答案，也就是 $\sum_{i=1}^{j-1} \frac{s_{ij}}{b_i b_j}$。先把 b_j提出来，那么只需要考虑 $\frac{s_{ij}}{b_i}$,下面我们来研究 s_{ij} 的性质。

首先观察到 s_{ij} 对应的图形有四种情况：交出来为空，梯形，三角形，整个矩形，具体对应那种情况跟 l_j, r_j 有关。考虑把它拆成两个图形面积的差：即由 $x = l_i, x = r_i, y = l_j, y = x$ 围成的面积减去 $x = l_i, x = r_i, y = r_j, y = x$ 围成的面积。以上两个图形在给定 i 后面积只与 l_j, r_j 的取值有关，具体的说是一个关于 l_j（或 r_j）的不超过二次的多项式。

题解

我们有：

$$f(x) = \begin{cases} 0 & x > r_i, \\ \frac{(r_i - x)^2}{2b_i} & l_i < x \le r_i, \\ -x + \frac{l_i + r_i}{2} & x \le l_i. \end{cases}$$

题解

我们有：

$$f(x) = \begin{cases} 0 & x > r_i, \\ \frac{(r_i - x)^2}{2b_i} & l_i < x \le r_i, \\ -x + \frac{l_i + r_i}{2} & x \le l_i. \end{cases}$$

多个 i 的贡献可以通过多项式系数直接相加得到，用树状数组维护系数的前缀和即可，离散化后可以做到 $O(n \log n)$。对于 b 有 0 的情况，本质上是一样的，矩形退化为线段，面积退化为长度，用同样的方式处理即可。

Paint the Graph

给一张连通图，每个点上有一个点权和颜色（黑白）。每次可以选择一条边，交换两点点权，同时若两点颜色相同则两点颜色均取反，否则不变；问是否可以通过交换从初始状态变成最终状态。

题解

其实操作等价于让数字带上颜色，每次相当于把两个相邻点上的数字交换，然后两个数字上的颜色都发生改变。

如果数字集不同，那么肯定为 No. 因为是连通图，所以如果不管颜色，每个数字必定能够到达指定的位置。

现在加上颜色，如果是二分图的话，那么对于任意一个数字，它在二分图任意一边的时候它所带的颜色都是固定的。对于二分图左右染色，我们记数字的新颜色为：它本身带的颜色 $\oplus$ 这个数字在二分图上的颜色，发现无论怎么变换，数字的"新颜色"都不会改变，只需判断两种新颜色所对应的数字集前后相同即可。

题解

对于一般图，我们可以看成二分图的基础上有若干奇环，奇环的作用在于，可以把两个数字拉到奇环上，然后进行交换，此时两个数字的“新颜色”都异或 1. 因此对于有奇环的情况，只需要前后状态所有数字 “新颜色” 异或和相同且总数字集相同即可。

复杂度瓶颈在判断数字集是否相同，为 $O(n \log n + m)$.

Yazid and Quiz Show

给定 n 个大整数集合，在每个数字集合中等概率随机选取一个数，并将得到的 n 个数按照字典序最大顺序拼起来，求结果的期望。

数字总个数个数$\leq 5 \times 10^5$，数字总位数$\leq 2 \times 10^6$；

题解

首先考虑给定数字，求拼接的最大字典序结果：

结论为存在全序关系 $(\succ, \mathbb{Z}^+)$，定义为 $[a \succ b] \equiv [a + b > b + a]$（后面的+表示字符串意义的拼接，>表示字典序比较），容易证明最终结果可按上述全序关系排序得到。

题解

首先考虑给定数字，求拼接的最大字典序结果：

结论为存在全序关系 $(\succ, \mathbb{Z}^+)$，定义为 $[a \succ b] \equiv [a + b > b + a]$（后面的+表示字符串意义的拼接，>表示字典序比较），容易证明最终结果可按上述全序关系排序得到。

上述排序可以利用 Hash + 二分 做到 $O\left(n \log\left(\sum k_i\right) \max\left\lceil \lg\left(S_{i,j}\right)\right\rceil\right)$。

题解

接着考虑计算贡献：对于数字 v，假设其来自 i_v

设 $g_{i,j}(j \geq 1)$ 表示 $|\{x \in B_i \mid x \preccurlyeq v, \mathrm{len}_x = j\}|$，$g_{i,0} = |\{x \in B_i \mid x \succ v\}|$

设 $f_i(x) = \sum_j g_{i,j} x^j$，则其贡献为

$$v \sum_i 10^i \left([x^i] \prod_{k \neq i_v} f_k(x) \right)$$

题解

接着考虑计算贡献：对于数字 v，假设其来自 i_v

设 $g_{i,j}(j \geq 1)$ 表示 $|\{x \in B_i \mid x \preccurlyeq v, \mathrm{len}_x = j\}|$，$g_{i,0} = |\{x \in B_i \mid x \succ v\}|$

设$f_i(x) = \sum_j g_{i,j} x^j$，则其贡献为

$$v \sum_i 10^i \left([x^i] \prod_{k \neq i_v} f_k(x) \right)$$

问题在我们按顺序处理v的前提下变为全局乘积和单点修改，可以线性处理，复杂度瓶颈在排序。

题解

接着考虑计算贡献：对于数字 v，假设其来自 i_v

设 $g_{i,j}(j \geq 1)$ 表示 $|\{x \in B_i \mid x \preccurlyeq v, \text{len}_x = j\}|$，$g_{i,0} = |\{x \in B_i \mid x \succ v\}|$

设$f_i(x) = \sum_j g_{i,j} x^j$，则其贡献为

$$v \sum_i 10^i \left([x^i] \prod_{k \neq i_v} f_k(x) \right)$$

问题在我们按顺序处理v的前提下变为全局乘积和单点修改，可以线性处理，复杂度瓶颈在排序。

注意特判 0。